Reorganization of physical similarities in source code using clustering

Muhammad Kashif Siddique Randhawa
mkashif173@gmail.com
Department of Computer Science
University of Agriculture Faisalabad, Pakistan

Dr. Imran Mumtaz
imranmumtaz@uaf.edu.pk
(Asst. Professor) Department of Computer Science
University of Agriculture Faisalabad, Pakistan

Abstract— The objective of research is to uncover the usage of clustering with other source code detection techniques(Semantic, Structure, Kernel based, Adaptive Local Alignment and Graph based) to find out efficient physical source code similarity. Clustering is typically methodology to develop the gatherings of related items that is utilized for group the data. Source Code Similarity techniques are less efficient and slow in some situations. For find the clustering efficiency with detection techniques, I have conduct a survey in which make the comparison and shows which similarity techniques are best to use. To sort out the problem which can be effective for find efficient source code similarity technique, I have conducted a research. The research is basically based on the Survey which technique is used best for source code plagiarism. For taking the results use the SPSS statistical tool. . By using the clustering technique with other detection techniques will give more accurate and exact results in the physical similarity of source code. The results obtained are effective in efficiency and accuracy.

Index Terms— Clustering, Physical Similarities, Plagiarism, Source Code, Similarity Techniques, Detection Techniques, Code Similarity.

1 INTRODUCTION

To Identify and reorganize the similarity and difference amount data is main objective of computer science. To find out the similarity among source code is linked up with many interested areas of computer science like academic projects, assignments and qualitative software development. In this research, we focus the methodology which can find out the best similarity among source code using the clustering. The basic purpose is to get out the best similarity methods which can give the best result by combining with others source code detection techniques.

Clustering algorithms divide the set of objects into groups called the clusters as same homogenous groups of similarity and dissimilarity. For much application more than one matrix abstained through clustering and sum up matrix for find the similarity and dissimilarity groups. Clustering algorithm find out the object groups called cluster. Clustering is used in application of natural science, engineering, economics, computer science and other fields (Santi *et al.,* 2016).

1.1 Introduction of Origin Code and its similarity recognition methods

Source code is group of statements, given to computer for perform instruction. Language which used for writing source code has a syntax which compiled into machine language for computer processing. Clustering algorithms divide the set of objects into groups called the clusters as same homogenous groups of similarity and dissimilarity. For much application more than one matrix abstained through clustering and sum up matrix for find the similarity and dissimilarity groups (Mubashar *et al.*, 2013).

1.2 Types of Clustering

Diverse Types of Clustering utilized for complementing source code likeness are clarified as tails one specific.

1.2. 1 Physical Clustering

Physical grouping will be found with thing code similarity and match includes substance. Physical similarity exists if join code exist in both associations. It can be associated by organizing groups in records. (Mubashar *et al.*, 2013).

```
Public Student(String FN, String LN)
{
FName = FN;
LName = LN;
Cnt ++;
System.out.printf(" Student Constructor: %-30s%\t Cnt = %d\n" FName,LName,Cnt);
}
```

1.2.2 Conceptual Clustering

Physical code might be different in elements in any case they create the comparative result.

Code Segment 1	Code Segment 2

int a=10;
int b=15;
int c=(a+b)*2;
	int ab=10;
	int ba=15;
	int cd=(a+b)*2

1. 2. 3 Hierarchical Clustering

Progressive division type of clustering, a particular endpoint is achieved, and the area of every item in the pecking order bunch collection consolidates clean these groups increment. The utilization of an arrangement all through, each protesting the bunch, or split the gathering into littler gatherings, the various leveled area, the end condition is content with their own particular end condition.

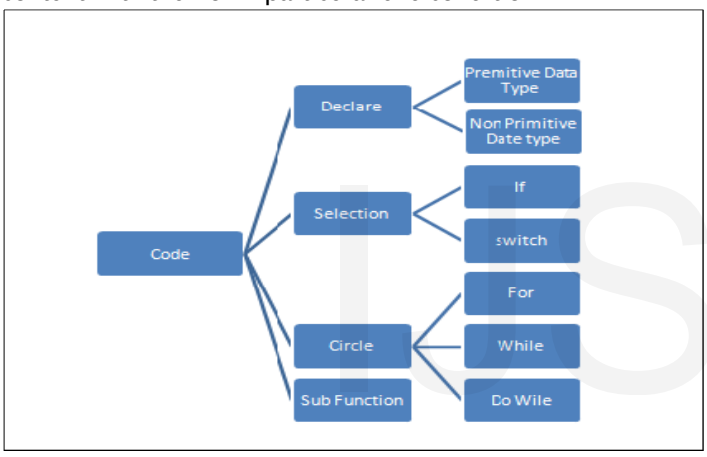

Fig. 1.1

1.2.5 K-mean grouping

K-implies procedure isolates the information into the gatherings, which is best technique for parceling group utilizing the model. Framework is organized bases which ignores adjust variable names. Getting the literary theft distinctive apparatuses are utilizing taking into account the kind of calculations. Like as in unique mark framework use coordinating calculation that match watchwords (Shamir *et al.,* 2013)

1.3 Similarity Detection utilizing nearby Alignment

1.3.1 Local Alignment

Local arrangement is ordinarily known as the Smith-Waterman calculation. Smith-Waterman calculation was made with a specific end goal to discover areas of similitude between two nucleotide or protein groupings to the starting point. To decide the individual report relating closeness lattice, position of the premises, to a few sub-system to support the coordinating score of the arrangement region (Ji *et al.,* 2007).

1. 3. 2 Adaptive Local Alignments

The basic strategy of the adaptive local alignment is that the matching score of keyword should reflect the frequencies of keywords. More specifically, we attributed the matching score of keywords in reverse of the frequencies of them Since phenomenal to get a low repeat watchword used by both undertakings meanwhile, it is two tasks should be seen as essentially the same as the use of catchphrase repeat set low (Ji *et al.,* 2007).

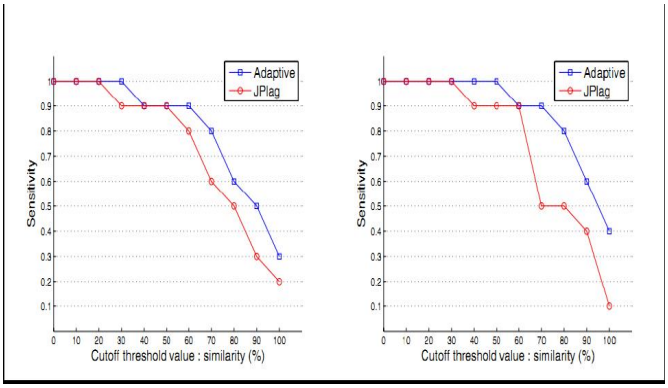

Fig-1.2 Evaluation of Adaptive tools with JPlag

1.3.3 Fuzzy Clustering

Fuzzy clustering methodologies are a reasonable answer for distinguishing source-code written falsification because of their ability to catch the subjective and semantic components of comparability. Execution of the methodology is contrasted with the best in class written falsification location Running Karp-Rabin Greedy-String-Tiling (RKR-GST) calculation. It depends on Fuzzy C-Means furthermore, the Adaptive-Neuro Fuzzy Inference System (ANFIS) Georgina (Cosma, G., & M. Joy, 2012).

1.3.4 Tokenization

The token-based structure-metric approach to detect code similarity across several code trees has seen substantial research in the past. Token sequences can be seen as strings, and so recognizing code similitude is a type of finding the Longest Common Subsequence (LCS) between two strings. If an area of source code in info is moved, the calculations distinguish this as various contrasts of a solitary change in the code (Toomey, 2013).

1.3.5 Structure Metrics

The structure-metric methodology based on the lexical investigation. Every code base to be analyzed is first lexically investigated to deliver an arrangement of tokens. These token successions are then contrasted with find basic token subsequences which show similitude's between the code bases. Plot another way to deal with the structure-metric methodology which utilizes hashes of token strings. The advantages of this methodology over the current examination incorporate the synchronous correlation of various expansive code bases, quick execution and the fare of serialized token streams for delicate code bases (Toomey, 2013).

1.3.6 Latent Semantic

Latent Semantic Analysis is a data recovery method involving numerical calculations that are connected to text collections. At first a text collection is preprocessed and spoken to as a term-by-record grid containing terms and their recurrence checks in documents. Matrix changes are connected such that the estimations of terms in records are balanced contingent upon how as often as possible they show up inside and over records in the collection (Cosma & Joy, 2012).

1.3.7 Call Graph

One possible representation for the function-level flow is a function-call graph which represents dependencies among functions within a program. Program source codes are written with object-oriented concepts and several refactoring techniques, so that the codes are getting more and more modularized at functional level. Since a source code encodes program logic to solve a problem, the execution flow at function level is one of the important factors to identify the source code. Therefore, this function-level flow should be considered to compare source codes (Song *et al.,* 2015).

1.3.7 Kernel based

Algorithm utilized is K-implies which is an effective partition grouping procedure. In essential K-implies grouping calculation the two primary parameters are the quantity of parameters (K) and the underlying group focuses. Select K, where "K" no. of centroids is to be chosen. Assign out every item to the gathering that has the nearest centroid utilizing some separation or comparability measure. When the sum total of what items have been relegated, recalculate the positions of the "K" centroids. Repeat Steps 2 and 3 until the centroids no more change (Shakhovska & Shvorob, 2015).

2 REVIEW OF LITERATURE:

Juricic, (2012) proposed detail on detecting of source code Likeness utilizing Low-level Languages. Their cardstock proposed a reaction for perceive compulsion of initiation statute as to finding the goodness similarity identifying with the starting stage records. They displayed the goodness pluses and minuses from the strategy each and every through evaluate and figure out how to enhance the calculation criteria to raise its comfort close by unfaltering quality.

Cosma and Joy, (2012) described LSA-develop system depends regarding the corpus itself and on the choice of parameters which are not normally adaptable. Besides, a LSA-based system can't be evaluated by strategy for whether it can recognize specific copyright encroachment strikes in light of its dependence on the corpus, and this makes it difficult to differentiate. JPlag similarly has the weakness of not having the ability to fuse records that don't parse in the relationship strategy.

Shamir *et al.,* (2013) Focused detail on another way to deal with the discovery of Source code similitude, the structure-metric methodology which utilizes lexical examination. The advantages of this methodology over the current examination incorporate the synchronous correlation of numerous substantial code bases, quick execution and the fare of serialized token streams for delicate code. The token-based structure-metric way to deal with identify code similitude over a few code trees has seen generous examination previously. With our instrument take a novel methodology which goes astray from the customary utilization of postfix trees and Longest Common Subsequence varieties.

Shakhovska & Shvorob, (2015) proposed two algorithms for locating and weird duplicate were considered. The verification of the considered algorithms and merged algorithm was made. System analysis for the intelligent system of determines the degree of resemblance of the texts was transported out and two charts were developed. For instance, using the method of "descriptive words “can determine what type includes documents are scanned as each produced vector distinctly identifies this class. Then simply identify duplicates in a particular class of documents, validations using methods based on the analysis of special similarities.

Acampora & Cosma, (2015) proposed Source-code blandness disclosure in coding, concerns the ID of source-code documents which contain for all intents and purposes indistinguishable and/or misty source-code pieces. Delicate social affair techniques are a legitimate reaction for perceiving source-code consistency needed to their essentialness to record the subjective and semantic parts of likeness. In light of current circumstances, Fuzzy packaging approaches have never been overviewed on the source-code formed corruption territory issue as much as different methods.

Santi *et al.,* (2015) presented cluster information gathered from heterogeneous uniqueness frameworks. The model at the same time assigns people to associations with comparative bunching and, for each and every gathering,

decides the best grouping arrangement. With the possibility to do as such in light of the fact that the nearby hunt dispatch depends on three neighborhood plunges installed into the Variable system.

3 RESEARCH AND METHADOLOGY

To understand and research the best similarity method conducts the survey to find out the behavior of different plagiarism methods. Asking a question and gathering information as words that is broke down and scanning for topic. The quantitative data collection methods based on random sampling and structured data collection techniques that fit different experiences into predetermined response categories.

A survey consists of questions aimed at extracting specific data from a particular group of people. Survey contains predetermined set of questions that is given to a sample. It allows generalizing the findings from the sample to the population, which is the whole purpose of survey research.

Table 3.1:

Questionnaire:

1. Local Adaptive Technique matched keywords among different source files is highly effective.
2. Clustering is data mining technique is more popular than other approaches now days
3. Due to plagiarism issues, is it difficult to apply efficient plagiarism detection technique?
4. All Detection tools are efficient for each type source code.
5. It is easy to integrate semantic method with clustering in software systems without error?
6. Using Token string matching, tree and Structure-based techniques can be applicable on all languages?
7. Pure tree- and graph-based code comparison techniques are cost effective then other techniques?
8. Structure metrics and adaptive local alignments detect high similarity in different LOC source codes?
9. Do you think that Clustering is efficient for your effort in identifying and analyses the high source code similarity?
10. The cost efficient of plagiarism detection methods are high time expensive
11. Difficulties in source code plagiarism analysis and testing problems?
12. Detection methods are language-dependent and can it be applicable on all languages
13. Kernel-based computation (string, sub tree) is efficient and not expensive than other techniques
14. Latent semantic technique with clustering can give high similarity and accurate results for source code
15. Control Structures technique find highly plagiarism than other techniques
16. Different plagiarism detection techniques used together can find out high level of similarity
17. Tokenization and Comparison Algorithm combine can give effective plagiarism in source code files
18. Recent detection system is a structure-based system efficient for change elements such as variable names and other comments empty.
19. Combing the Weibull and local adaptive alignment techniques can give the high similarity result
20. After applying the combined plagiarism techniques with clustering, source code similarity will be high?

3.1. Purpose of the survey:

To find out the best similarity method in source code similarity, conduced a web-based survey which consists of 20 questions on source code plagiarism. From the investigation displayed general regions of particular area to source code plagiarism. Out final category was put into include which technique will be better for effective plagiarism detection.

3.2. Statistics used in base software

Descriptive statistics used as Cross tabulation, Frequencies, Descriptive, Explore, Descriptive Statistics. SPSS tools are used for statistical analysis of survey results. It concludes the cross tables and graphs which show the results after applying statistical analysis on primary data.

4 RESULTS AND DISCUSSION

An electronic overview was made on the Google forms. The web-join for the study was circulated to a rundown of scholastics and to software organizations. The mailing list comprised of 60 names, large portions of whom can be expected to have ability in showing programming and programming analyses. The study was mysterious, however incorporated an area in which the association could alternatively give individual data. The survey contained for the most closed questions requiring different decision reactions. The inquiries were as little situations portraying different ways got, utilized, and recognized material. The respondents were required to choose from a decision of reactions the kind of pertinent answer that as they would like to think connected to every situation. From the investigation displayed general regions of particular area to source code plagiarism techniques. Out final category was put into include which technique will be better idea for effective plagiarism detection method. Statistical analysis and their full report is available.

4.1 .Survey Results

In the organization many respondents of underlying study end study were the same. This creates the probability to analysis the results with measurable techniques. The results conclude from the study and show in diagrams.

4.1.2. Initial and end survey

Initial and survey based on quantitative analysis which statistical analysis of clustering techniques effectiveness in software field and academic field. Test compares the association among the variables.

4.2. STATISTICAL ANAALYSIS

Information gather from the respondent's (Question 2) are displayed in Table 4.1. In which respondents (74%) agreed with the effect of clustering in source code similarty and rest either (36%) respond in the Neutral way.

Table 4.1:

Clustering Most popular	Respondents (%)
Agree	74.7
Natural	16.2

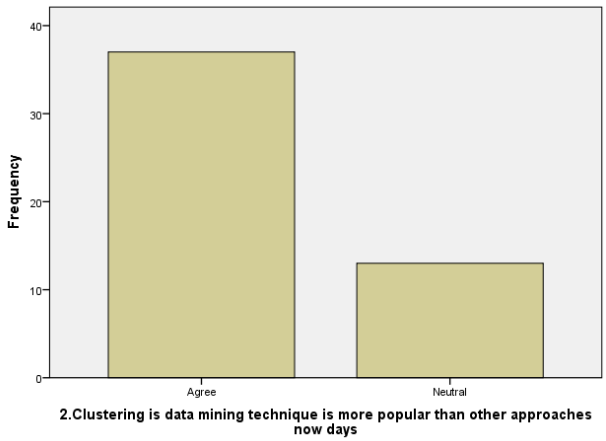

Fig.4.1

Considering Question 3 (Table 4.2), a large portion of the respondents (88.8%) agree that dua to plagiarism technique issue difficult to apply efficient technique, while 15.4% no respond in against scenario.

Table 4.2:

Diffictuly in selection most Effective technique	Respondents (%)
Agree	80.8
Natural	15.4

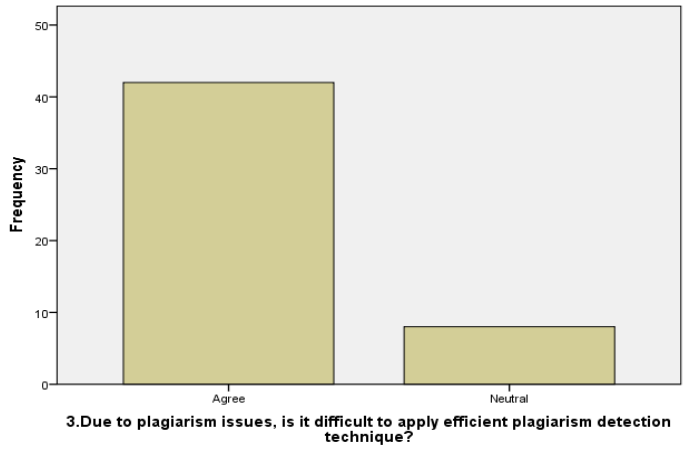

Fig.4.2

Repsond from Question 5 (Table 4.3), a large portion of the respondents (76.9%) respond in agree that Semantic method can be integrate with clustering technique for better results in source code similarty and other (19.2%) respond in natural against scenario.

Table 4.3:

Semantic Technique with Clustering	Respondents (%)
Agree	76.9
Natural	19.2

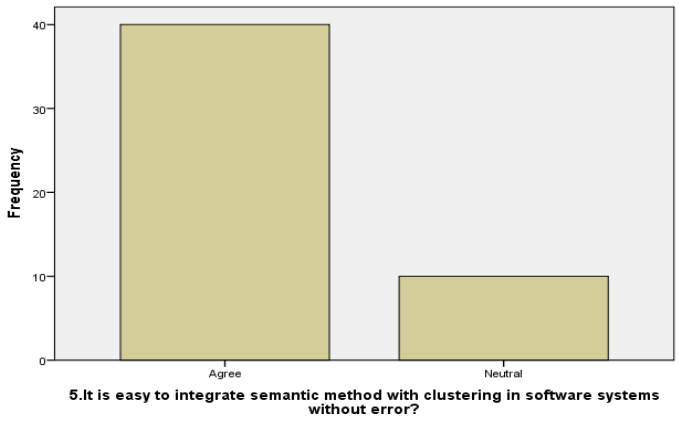

Fig.4.3

Considering Question 9 (Table 4.4), a portion of the respondents (44%) agree, 24% disagree with clutering, 16% reppond in natural. That's mean the clustering is efficient in most satiation integrate with other detection technique.

Table 4.4:

Clutering is efficient in analys similarty	Respondents (%)
Agree	44
Natural	16
Disagree	24
Strongle Disagree	16

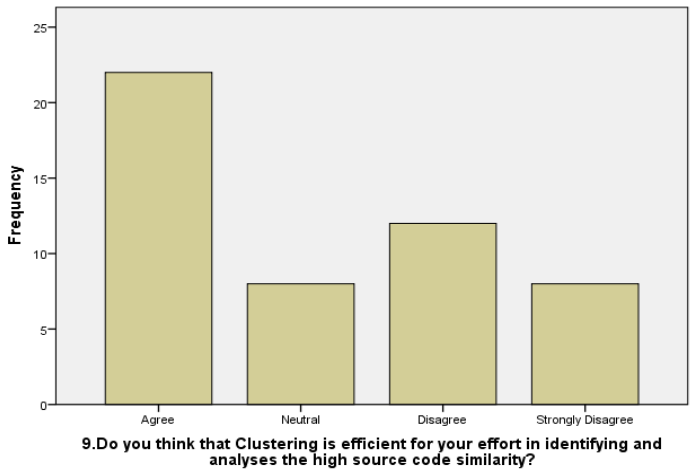

Fig.4.4

Considering Question 16 (Table 4.5), 38% respondand answer in the agree that differenct plagiarism detection techniques two or more after combine can find high level of

source code similarty while 10%disagree with effectivnes of detection techniques after combinig and other 52% respond in natural.

Table 4.5:

Different Detection Technique can combine	Respondents (%)
Agree	38
Natural	52
Disagree	10

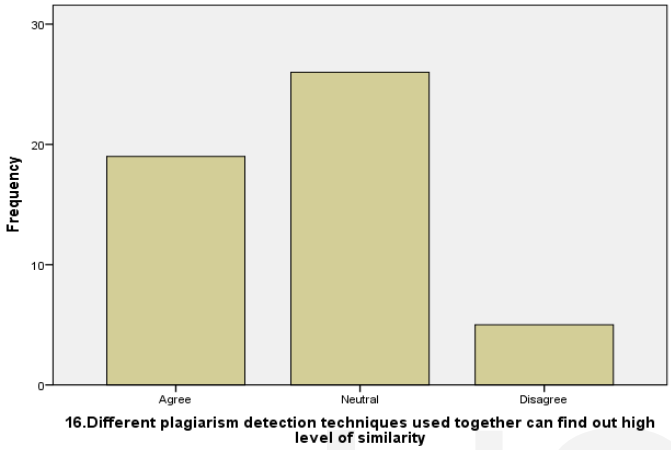

Fig.4.5

Considering Question 19 (Table 4.6), 76% respondand answer in the agree that combining the Weibull distribution and local adaptive alignment can give high similarty results with using the clustering technique in efficiency and accuracy, while 4%disagree with comination of detection techniques and other 20% respond in natural.

Table 4.6:

Weibull and Local Adaptive alignment Technique together high effective	Respondents (%)
Agree	76
Natural	20
Disagree	4

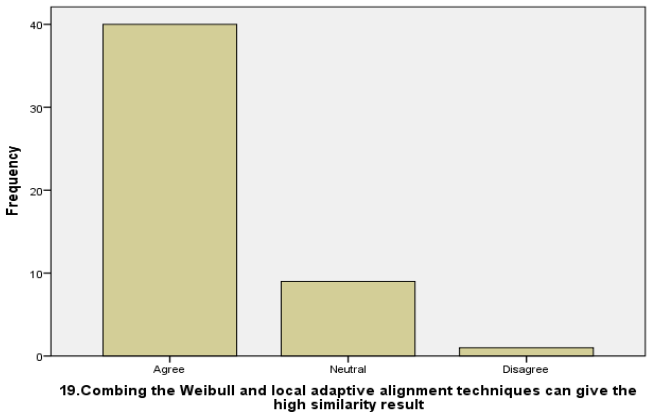

Fig.4.6

As result of question 20 (Table 4.7), the plagiarism technique with the clustering can give better result for detection the source code similar in which (84) % respond and strongly agree, while other 4% disagree or no response.

Table 4.7:

Applying Detection technique with clustering	Respondents (%)
Agree	84
Natural	12
Disagree	4

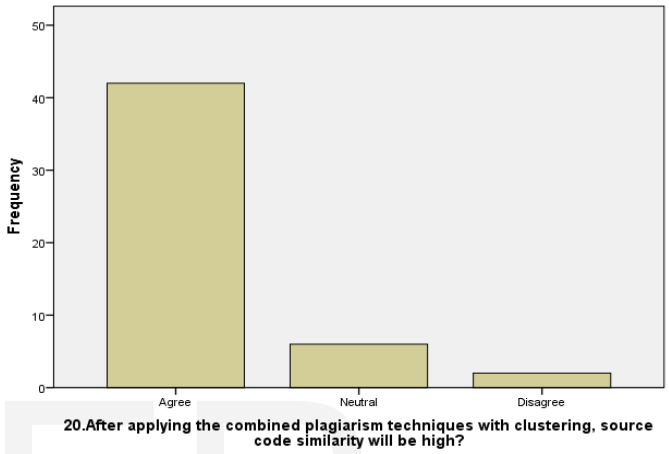

Fig 4.7

Figure 4.7 Results shows that the combining the source code detection techniques using crusting can be better for enhance similarities and remove the other draw backs.

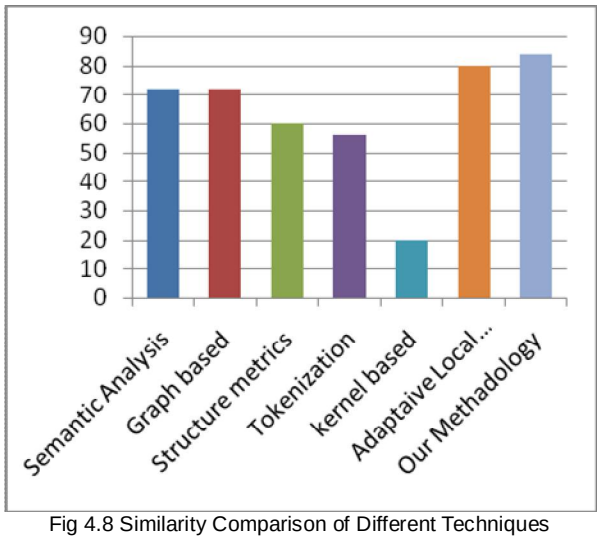

Fig 4.8 Similarity Comparison of Different Techniques

Similarity comparison diagram shows that clustering significance better perform than others methodologies. It combines the both effect of keywords matching, structure metric and same time the expressions matching's.

Conclusion

My Proposed analysis gives efficient results than other source code similarity detection techniques. Other detection techniques distinctly can don't generate the efficient source code similarity. Previous Techniques are cost effect and time effect which cannot give effective similarity results. Clustering technique is used for cluster the similar objects. By using the data mining technique of clustering which combine the similar objects in a group which give the effective results in software modification. By conducting survey about different techniques statistically analyzed that other detection techniques with the clustering can enhance in find out source code similar more accurately. It will give efficient result and save the time for software modification and in academic instituted of greater source code similarity. This survey result concludes that physical clustering technique for the efficient source code is better working combine with all other methods. Combine techniques is not the end of project. In future adopt the more methods for better performance. Automatic system can be added which keeps the system more efficient. Further we can minimize the time cost effort.

REFERENCES

[1] Acampora, G., and G. Cosma, 2015, August. A Fuzzy-based approach to programming language independent source-code plagiarism detection. *In Fuzzy Systems (-IEEE), 2015 IEEE International Conference on* IEEE,1-8. FUZZ

[2] Cosma, G., and M. Joy, (2012). An approach to source-code plagiarism detection and investigation using latent semantic analysis. *IEEE transactionson computers*, 61(3), 379-394.

[3] Ji, J., S. H. Park, G., Woo, and H. G. Cho, (2007, December). Source code similarity detection using adaptive local alignment of keywords. *In Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007)* (pp. 179-180). IEEE.

[4] Juricic, V. 2011. Detecting source code similarity using low-level languages. *In Information Technology Interfaces (ITI), Proceedings of the ITI 2011 33rd International Conference on* IEEE, 597-602.

[5] Mubashar ur Rehman, M. Nadim Asif, R. Talib, M. U. Sarwar ,B. Ali (2013) Identification of Physical Similarities in Source Code, *International Journal of Computer Science and Management Research,* Vol 2 Issue 5 May edn.

[6] Santi, É., D. Aloise, and S. J. Blanchard, (2016). A model for clustering data from heterogeneous dissimilarities. *European journal of Operational Research*, 253(3), 659-672.

[7] Song, H. J. et al., (2015). Computation of Program Source Code Similarity by Composition of Parse Tree and Call Graph. *Mathematical Problemsin Engineering*, 2015.

[8] Shamir, L., J. F. Wallin, A. Allen, B. Berriman, P. Teuben, R. J. Nemiroff, and K. DuPrie, 2013. Practices in source code sharing in astrophysics. *Astronomy and Computing,* 1(1): 54 58.

[9] Dr. Warren Toomey, Code Similarity Detection in Multiple Large Source Trees using Token Hashes", *PAN-09 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse and 1st International Competition on Plagiarism Detection,* 2010.

[10] Shakhovska, N., and I. Shvorob, (2015, September). The method for detecting plagiarism in a collection of documents. *In Scientific and Technical Conference" Computer Sciences and Information Technologies*"(CSIT), 2015Xth International (pp. 142-145). IEEE.

[11] Vani, K., and D. Gupta, (2014, November). Using K-means cluster based techniques in external plagiarism detection. *In Contemporary Computing and Informatics (IC3I), 2014 International Conference on* (pp. 1268-1273). IEEE.